



TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií



Softwarová detekce polohy leteckého profilu v obrazu

Bakalářská práce

Studijní program: B2646 – Informační technologie
Studijní obor: 1802R007 – Informační technologie
Autor práce: **Ondřej Jakub**
Vedoucí práce: Ing. Petr Ječmen
Konzultant: doc. Ing. Petr Šidlof, Ph.D.





TECHNICAL UNIVERSITY OF LIBEREC
Faculty of Mechatronics, Informatics
and Interdisciplinary Studies ■

Software detection of the position of an airfoil in the image

Bachelor thesis

Study programme: B2646 – Information Technology
Study branch: 1802R007 – Information Technology
Author: **Ondřej Jakub**
Supervisor: Ing. Petr Ječmen
Consultant: doc. Ing. Petr Šidlof, Ph.D.



ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Ondřej Jakub**
Osobní číslo: **M13000106**
Studijní program: **B2646 Informační technologie**
Studijní obor: **Informační technologie**
Název tématu: **Softwarová detekce polohy leteckého profilu v obrazu**
Zadávající katedra: **Ústav nových technologií a aplikované informatiky**

Z á s a d y p r o v y p r a c o v á n í :


1. Seznamte se s problematikou (viz. [1], [2])
2. Prostudujte algoritmy pro detekci objektu v obrazu, možnostmi segmentace objektů v obraze a způsoby detekce významných bodů
3. Seznamte se s knihovnou OpenCV, její strukturou a možnostmi použití v jazyce C#
4. Vytvořte aplikaci realizující detekci souřadnic náběžné a odtokové hrany

Rozsah grafických prací: dle potřeby
Rozsah pracovní zprávy: 40 - 60 stran
Forma zpracování bakalářské práce: tištěná/elektronická
Seznam odborné literatury:

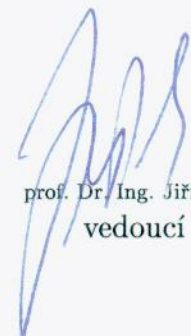
- [1] Šidlof P., Vlček V., Štěpán M., Horáček J., Luxa M. et al. (2014), Wind tunnel measurements of flow-induced vibration of a NACA0015 airfoil model, in: Proceedings of the ASME 2014 PVP Conference - symposium Fluid-structure interaction, Anaheim, CA, USA
[2] Štěpán M., Šidlof P., Vlček V. (2013), Měřicí řetězec aerodynamického profilu se dvěma stupni volnosti, in: Interactions and feedbacks 2013, Prague, Czech Republic
[3] Tropea C., Yarin A. L., (Eds.) J. F. F. (2007), Springer Handbook of Experimental Fluid Mechanics, Springer.
[4] Shapiro A. H. (1954), The Dynamics and Thermodynamics of Compressible Fluid Flow, The Ronald Press Company, New York.
[5] Šonka V., Hlaváč M., Boyle R. (2014), Image Processing, Analysis, and Machine Vision, 4th edition
[6] Albahari J., Albahari B. (2012), C# 5.0 in a Nutshell: The Definitive Reference, 5th Edition

Vedoucí bakalářské práce: **Ing. Petr Ječmen**
Ústav nových technologií a aplikované informatiky
Konzultant bakalářské práce: **doc. Ing. Petr Šidlof, Ph.D.**
Ústav nových technologií a aplikované informatiky

Datum zadání bakalářské práce: **20. října 2015**
Termín odevzdání bakalářské práce: **16. května 2016**


prof. Ing. Václav Kopecký, CSc.
děkan




prof. Dr. Ing. Jiří Maryška, CSc.
vedoucí ústavu

V Liberci dne 20. října 2015

Prohlášení

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

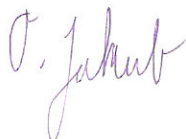
Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum: 13.5.2016

Podpis:



Poděkování

Rád bych poděkoval panu Ing. Petru Ječmenovi za trpělivé vedení mé práce a odborný dohled. Děkuji také Mrg. Ivaně Jakobové a Bc. Barboře Urbancové za pomoc při gramatické a jazykové kontrole práce.

Abstrakt

V této práci je popsán postup, jakým jsou zpracovány snímky z měření v Novém Kníně, kde se měří proudění kolem leteckého profilu. Snímky použité v práci byly získány při měřeních na leteckém profilu NACA, a poté zpracovávány v programu IFGPro. Pro realizaci práce byl použit jazyk C# a jeho rozšiřující knihovna OpenCV. První část práce je úvodem do problematiky zpracování obrazových materiálů, další část se zabývá různými postupy při vyhodnocování snímků. Nakonec je popsána nejvhodnější metoda zpracování obrazu z hlediska přesnosti a časové náročnosti pro tento typ snímků.

Klíčová slova

NACA profil, flutter, interferometrické měření, EmguCV, zpracování obrazu

Abstract

In this work is described the process of how images are processed from measurements in Novy Knin, which measures the flow around an airfoil. The images used in the work were obtained during measurements on the NACA airfoil and are then processed in the program IFGPro. For the realization of the work was used C# language and its extension library EmguCV. The first part of work is an introduction to image processing, the next part is focused on different procedures for image processing. Last part, is describes best image processing method in terms of accuracy and time required for this type of images.

Keyword

NACA airfoil, flutter, interferometric measurement, EmguCV, Image processing

Obsah

Prohlášení.....	4
Poděkování.....	6
Abstrakt.....	7
Seznam symbolů, zkratk a termínů.....	10
Úvod.....	11
Profil NACA.....	11
1 Metody a funkce pro zpracování obrazu.....	13
1.1 Segmentace obrazu.....	13
1.1.1 Prahování.....	13
1.2 Detektory hran.....	14
1.2.1 Maticové hranové detektory.....	14
1.2.2 Konvoluce	14
1.2.3 Robertsův hranový detektor.....	15
1.2.4 Prewittův hranový detektor.....	15
1.2.5 Sobelův hranový detektor.....	16
1.2.6 Cannyho hranový detektor.....	16
1.3 Sledování hranice.....	17
1.3.1 Hystereze.....	17
1.3.2 Connected-component labeling.....	17
1.4 Filtrace šumu v obraze.....	17
1.4.1 Průměrování.....	17
1.4.2 Mediánový filtr.....	18
1.4.3 Lineární filtr.....	18
1.4.4 Gaussův filtr.....	19
2 Použité prostředky.....	20
2.1 IFGPro.....	20
2.2 Microsoft Visual Studio.....	20
2.3 Programovací jazyk C#.....	21
2.4 OpevCV.....	21
2.5 EmguCV.....	21

2.6 Snímky z měření.....	22
3 Návrhy řešení.....	23
3.1 Prahování.....	23
3.2 Hranové detektory.....	24
3.2.1 Maticové hranové detektory.....	24
3.2.2 Sobelův hranový detektor.....	25
3.2.3 Prewittův hranový detektor.....	26
3.2.4 Cannyho hranový detektor.....	26
4 Realizace vlastního řešení.....	28
4.1 Grafické rozhraní aplikace pro testování.....	28
4.2 Vyhlazení: rozmazání obrazu pro odstranění šumu.....	29
4.3 Nalezení gradientů a lokálních maxim.....	30
4.4 Eliminace nevýznamných hran – prahování.....	31
4.5 Sledování hranice hysterezí.....	32
4.6 Nalezení náběhové a odtokové hrany.....	33
4.7 Testování	35
Závěr.....	37
Seznam použité literatury.....	39

Seznam symbolů, zkratk a termínů

JPG	standardní metoda ztrátové komprese používané pro ukládání počítačových obrázků ve fotorealistické kvalitě
PNG	grafický formát určený pro bezztrátovou kompresi rastrové grafiky
C#	vysokoúrovňový objektově orientovaný programovací jazyk vyvinutý firmou Microsoft
GUI	neboli Graphical User Interface je uživatelské rozhraní, které umožňuje ovládat aplikaci pomocí interaktivních grafických ovládacích prvků.
NACA	National Advisory Committee for Aeronautics je federální úřad Spojených států založený na podporu leteckého výzkumu
IDE	Integrated Development Environment je software usnadňující práci programátorů zaměřený většinou na jeden programovací jazyk

Úvod

Hlavním cílem této práce je navrhnout a realizovat algoritmus, který by zefektivnil detekci polohy aerodynamického profilu NACA v obraze v daném okamžiku. Použité snímky jsou získávány při měřeních v Novém Kníně v rámci řešení projektu Grantové agentury ČR ve spolupráci s Ústavem termomechaniky Akademie věd ČR.

Při měřeních se sleduje nastartování a vývoj flutteru, tedy rozkmitání profilu v aerodynamickém tunelu a chování leteckého profilu. Na letecký profil působí v každém okamžiku vztlačová síla, odporová síla a moment síly. Pro získávání dat a snímků je použita vysokorychlostní kamera, série senzorů na leteckém profilu a Mach-Zenderův interferometr, který zachycuje změny v proudění a tlaku na leteckém profilu.

Interferometr využívá rozdílné průchodnosti světla skrz tekutiny s rozdílnou hustotou. Interferometr lze při měřeních změn tlaků nahradit metodou particle image velocimetry (PIV). Pomocí PIV lze také sledovat změny v tlaku a směru proudění. Interferometr je zásadní pro pozdější zpracování dat, neboť snímky z něj jsou použity v programu IFGPro. Jako průvodní materiály při zpracování bakalářské práce byly použity práce [1] a [2].

Při vyhodnocování v programu IFGPro se musí ručně nalézt letecký profil ve snímku, tedy náběhová a odtoková hrana. Tato realizace je značně časově neefektivní, neboť při měřeních je použita vysokorychlostní kamera a pořízených snímků jsou tisíce. Algoritmus by tedy měl nalézt náběhovou a odtokovou hranu v kratším čase než při ručním vyhodnocování. Snímky z interferometrických měření ve formátu JPG nebo PNG jsou obrazovým vstupem do vlastní realizace algoritmu. Po zpracování snímků jsou k dispozici přesné souřadnice náběhové a odtokové hrany v pixelech od levého horního rohu.

Profil NACA

NACA profily jsou Aerodynamické tvary křídla letadla, které zpracoval Národní poradní výbor pro letectví (NACA). Tvar profilů NACA je popsán za použití řady číslic za slovem "NACA".

Při měřeních je použit model křídla typu NACA0015 o délce 64,5 mm a šířce 80 mm, který je pružně uložen v rámu o rozměrech 80 x 210 mm. Model má dva stupně volnosti, rotační pohyb křídla zajišťuje torzní tyč umístěná v 1/3 třetiny křídla, zasunutá uvnitř profilu.¹

Rovnice pro výpočet profilu NACA:

$$Y_t = 5tc \left[0.2969 \sqrt{\frac{x}{c}} + (-0.1260) \frac{x}{c} + (-0.3516) \left(\frac{x}{c}\right)^2 + 0.2843 \left(\frac{x}{c}\right)^3 + (-0.1015) \left(\frac{x}{c}\right)^4 \right]$$

Obecná rovnice leteckého profilu

c	délka
x	pozice na třetině od 0 po c
Y _t	poloviční tloušťka na daném x
t	maximální šířka

¹ ŠIDLOF, Petr, Václav VLČEK, Martin ŠTĚPÁN, Jaromír HORÁČEK, Martin LUXA, David ŠIMURDA a Jan KOZÁNEK. Wind Tunnel Measurements of Flow-Induced Vibration of a NACA0015 Airfoil Model. In: Volume 4: Fluid-Structure Interaction [online]. ASME, 2014-7-20, V004T04A029- [cit. 2016-05-10]. DOI: 10.1115/PVP2014-28294. ISBN 978-0-7918-4601-8. Dostupné z: <http://proceedings.asmedigitalcollection.asme.org/proceeding.aspx?doi=10.1115/PVP2014-28294>. (Vlastní překlad).

1 Metody a funkce pro zpracování obrazu

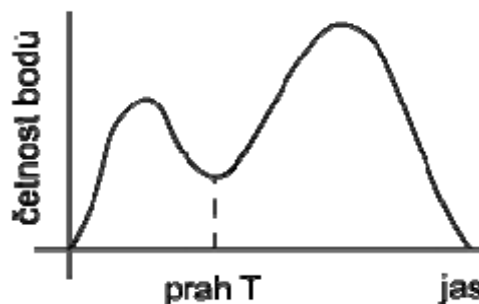
Zpracování obrazu je v informatice libovolná forma zpracování signálu pomocí matematických metod, jehož vstupem je video nebo sekvence snímků a výstupem je pak obraz a nebo množina parametrů a charakteristik. Většina metod pracuje s obrazem jako s dvourozměrným signálem.

1.1 Segmentace obrazu

Je skupina metod postavených na různých principech, které slouží k automatickému rozdělení vlastního obrazu na oblasti se společnými vlastnostmi a podobným významem. Výsledkem těchto operací je identifikace oblastí, které jsou ve scéně důležité.

1.1.1 Prahování

Jedná se o metodu, která omezuje barevný prostor a upravuje hodnoty vstupu podle předpisu a zvolených hodnot. Funkce je vhodná pro šedotónové obrázky, kdy na výstupu jsou pouze dvě barvy (bílá a černá). Tato funkce se velmi často používá jako příprava obrazu na pozdější zpracování, například detekci hran. Metody, které používají pouze jednu hodnotu prahu jsou vhodné pouze ve velmi jednoduchých aplikacích a proto mnohé složitější filtry a detektory používají alespoň 2 hodnoty pro prahování.²



Ilustrace 1: Funkce prahování

```
If Vstupní hodnota > Prahová hodnota:  
    Zobraz Max  
Else:  
    Zobraz Min
```

² Image processing, analysis, and machine vision. 4th edition. S.I.: Nelson Engineering, 2014. ISBN 1-133-59369-0. S. 179.

1.2 Detektory hran

Jsou mechanismy hledající oblasti v digitálních obrazech, kde se podstatně mění jas. Hrana je oblast v obraze - lze ji nalézt na rozhraní světla a stínu nebo na hranicích objektu – skoková hrana, nebo v místech trojrozměrných hran objektů – trojúhelníková hrana. Hrany jsou generovány na obou stranách detekce, hrana bývá velmi často zašumělá a snižuje kvalitu, proto je dobré šum odstranit nebo ho minimalizovat.³

1.2.1 Maticové hranové detektory

Při zpracování obrazu se používají matice pro rozmazání, ostření, detekci hran a další, čehož je docíleno prostřednictvím konvoluce mezi vstupním obrazem a konvolučním jádrem.

1.2.2 Konvoluce

Jedná se o matematickou operaci zpracovávající dvě funkce, jejíž výsledkem je funkce třetí. První funkcí konvoluce je souřadnicová síť pixelů, druhou funkci (konvoluční jádro) si lze představit jako matici, kterou systematicky přikládáme na příslušný pixel v obraze. Hodnotu jasu každého pixelu překrytého touto maticí vynásobíme hodnotou v příslušné buňce matice, sečteme a vynásobíme celkovým koeficientem konvoluční masky. Takto dostaneme novou hodnotu jasu pro každý pixel. Většinou používáme konvoluční masky s lichým počtem řádků a sloupců, protože potom leží reprezentativní pixel ve středu masky a můžeme dát středu větší váhu.⁴

Výsledný obraz závisí na jednotlivých prvcích konvoluční matice.

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Matice pro zostření snímku

$$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

Matice pro detekci hran

3 Image processing, analysis, and machine vision. 4th edition. S.I.: Nelson Engineering, 2014. ISBN 1-133-59369-0. S. 133-134.

4 Image processing, analysis, and machine vision. 4th edition. S.I.: Nelson Engineering, 2014. ISBN 1-133-59369-0. S. 53

1.2.3 Robertsův hranový detektor

Byl to jeden z prvních detektorů hran, jehož cílem je zjistit míru změny gradientu obrazu pomocí součtu rozdílů mezi diagonálně sousedními obrazovými body. Tento detektor je stále používán díky jeho rychlosti, ačkoliv díky malé velikosti matice je velmi citlivý na lokální šum.⁵

Velikost gradientu:

$$|G| = |f(x, y) - f(x+1, y+1)| + |f(x+1, y) - f(x, y+1)|$$

Výpočet gradientu Robertsova detektoru

Konvoluční matice:

$$\begin{array}{cccc} +1 & 0 & 0 & +1 \\ 0 & -1 & -1 & 0 \end{array}$$

Matice Robertsova detektoru

1.2.4 Prewittův hranový detektor

Robertsův detektor provede jednoduchý a rychlý výpočet 2D prostorového gradientu v obraze. Tím tak poukazuje na oblasti s vysokou prostorovou frekvencí, které často odpovídají hranám. Vstupem a výstupem je nejčastěji obraz ve stupních šedi. Tento detektor používá matice 3x3 pro detekci horizontálních a vertikálních hran a matice 5x5, která působí na větší ploše a je odolnější proti šumu.⁶

Matice 3x3:

$$G_x = \begin{array}{ccc} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{array} \times A$$

Matice pro detekci svislých hran

$$G_y = \begin{array}{ccc} -1 & -1 & -1 \\ 0 & 0 & 0 \\ +1 & +1 & +1 \end{array} \times A$$

Matice pro detekci horizontálních hran

Matice 5x5:

$$\begin{array}{ccccc} -2 & -1 & 0 & 1 & 2 \\ -2 & -1 & 0 & 1 & 2 \\ -2 & -1 & 0 & 1 & 2 \\ -2 & -1 & 0 & 1 & 2 \\ -2 & -1 & 0 & 1 & 2 \end{array}$$

Matice pro detekci svislých hran

^{5,6} Image processing, analysis, and machine vision. 4th edition. S.l.: Nelson Engineering, 2014. ISBN 1-133-59369-0. S. 137.

1.2.5 Sobelův hranový detektor

Sobelův hranový detektor provádí měření prostorového gradientu 2-D v obrázku, a tak klade důraz na oblasti s vysokou prostorovou frekvencí, které odpovídají hranám. Obvykle se používá k vyhledání přibližného absolutního gradientu v každém bodě obrazu, jehož vstupem je obraz ve stupních šedi. Tento detektor používá upravené matice Prewittova hranového detektoru pro přesnější detekci hran a poté pro potlačení šumu pomocí Gaussova filtru.

Upravené matice:

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \times A \quad G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} \times A$$

Upravené Prewittovy matice

Tyto matice lze upravit pro přesnější detekci hran v obraze.

1.2.6 Cannyho hranový detektor

V současné době je zřejmě nejlepším hranovým detektorem, a to kvůli několika krokům při jeho aplikaci. Mezi základní požadavky patří spolehlivá detekce (nalezení co nejvíce existujících hran), přesná lokalizace hrany (co nejpřesnější určení) a jednoznačnost (aby nebyly detekovány neexistující hrany).

Detektor pracuje v těchto krocích:

- 1) Eliminace šumu Gaussovým filterem (viz. Kapitola 1.4.4), aby nebyly detekovány nevýznamné detaily.
- 2) Určení velikosti a směru gradientu – tedy standartní detekce hran.
- 3) Nalezení lokálních maxim – výběr maximálních hodnot gradientu z předchozího kroku. Tím je zajištěno, že hrana bude v místě největšího gradientu.
- 1) Eliminace nevýznamných hran (prahování) – jsou odebrány nevýznamné hrany pomocí gradientu. Není vhodné určovat jeden gradient, protože hodnota může kolísat například kvůli šumu.⁷

⁷ Image processing, analysis, and machine vision. 4th edition. S.I.: Nelson Engineering, 2014. ISBN 1-133-59369-0. S. 144-145

1.3 Sledování hranice

Postup aplikovatelný převážně na obrazy s informací, jehož cílem je určení právě jen skutečné hrany a vyloučení artefaktů. Metoda může selhávat u příliš zašuměných obrazů a příliš komplikovaných tvarů.

1.3.1 Hystereze

Toto je poslední krok, kterým prochází například Cannyho hranový detektor. Hystereze znamená závislost stavu na stavech předchozích. V případě například Cannyho detektoru to znamená, že jsou ponechány pouze ty nejsilnější hrany a ty slabé, pokud nejsou spojeny s nějakou jinou silnější hranou, jsou potlačeny.⁸

1.3.2 Connected-component labeling

Tento postup se obvykle používá s hysterezí pro zjištění, které hrany budou zanechány. Jedná se o algoritmus, který vychází z teorie grafů. V případě Cannyho detektoru je tento krok použit pro spojování hran ve všech osmi směrech vedle silných hran. Pokud je alespoň v jednom sousedním bodě silná hrana, zůstane slabá hrana zachována, protože může být identifikována právě silnou hranou.⁹

1.4 Filtrace šumu v obraze

Vysoká úroveň šumu ve snímcích může omezit rozlišení detailů, snížit ostrost hran a poškodit vnímání ploch. Tato metoda může výrazně zlepšit celkový vzhled snímku. Většina metod pracuje na principu průměrování, což vede ke změkčení obrazu a problému s detekcí hran.

1.4.1 Průměrování

Je jedním z nejběžnějších filtrů, ale tento filtr ničí hranice objektů, což stěžuje jejich následnou identifikaci a přesnou detekci. Při použití toho filtru je obraz procházen po řádcích (princip konvoluce viz. Kapitola 1.2.2) a nová hodnota je spočítána jako průměr okolních hodnot za pomoci filtrační masky.


8 Image processing, analysis, and machine vision. 4th edition. S.I.: Nelson Engineering, 2014. ISBN 1-133-59369-0. S. 146.

9 Image processing, analysis, and machine vision. 4th edition. S.I.: Nelson Engineering, 2014. ISBN 1-133-59369-0. S. 197.

1.4.2 Mediánový filtr

Je další velmi často používaný filtr. Medián je číslo, které se po uspořádání nachází uprostřed posloupnosti. Jeho nalezení zjednoduší vhodně zvolená velikost zkoumaného okolí. Navíc není potřeba vytvořit posloupnost, ale stačí hodnoty částečně uspořádat. Jeho nevýhodou je, že poškozuje jemné čáry a ostré hrany v obraze.¹⁰

88	95	80
68	100	56
82	75	60



56, 60, 68, 75, **80**, 82, 88, 95, 100

Tabulka 1: Příklad použití mediánu

1.4.3 Lineární filtr

Platí pro něj princip superpozice a účelem této filtrace je obvykle potlačení nebo zvýraznění určitých spektrálních složek v obraze. Pro princip superpozice musí platit, že odezva lineárního systému tvořeného tímto filtrem na součet dvou či více signálů musí být rovna součtu odezev tohoto systému na jednotlivé signály.¹¹

$$F(ax_1 + bx_2) = aF(x_1) + bF(x_2)$$

Princip superpozice

Tento filtr lze na obraz aplikovat také pomocí metody konvoluce (popsané v kapitole 1.2.1), díky které je tento filtr velmi rychlý.

Mezi nejčastější dělení patří:

- dělení podle účelu
 - dolní propust (vhodná pro potlačení šumu v obraze)
 - horní propust
 - pásmová propust
 - pásmová zadrž
- dělení podle odezvy
 - filtr s konečnou impulzní odezvou (FIR)
 - filtr s nekonečnou impulzní odezvou (IIR)

¹⁰ Image processing, analysis, and machine vision. 4th edition. S.I.: Nelson Engineering, 2014. ISBN 1-133-59369-0. S. 130-133.

¹¹ Image processing, analysis, and machine vision. 4th edition. S.I.: Nelson Engineering, 2014. ISBN 1-133-59369-0. S. 51-52.

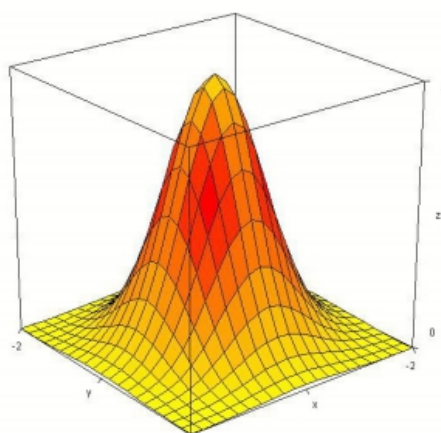
1.4.4 Gaussův filtr

Je to jedna z metod na odstranění šumu pomocí masky, která se skládá z elementů určených Gaussovou funkcí. Největším problémem při filtrování Gaussovým filterem je volba správného σ , které upravuje šířku odezvy a okolí ve kterém Gaussův filtr reaguje. Při zvolení příliš malého σ nebo filtrovaného okolí, je filtrace nevýrazná, ale zase nepoškozuje hrany. Při zvolení vyšších hodnot σ a filtrovaného okolí je filtrace výrazná, ale lokalizace hran přestává být přesná, popřípadě některé méně výrazné hrany zaniknou úplně.¹²

2D Gaussova funkce:

$$G(x, y) = \frac{1}{2\pi\sigma^2} \cdot e^{\frac{-x^2 - y^2}{2\sigma^2}}$$

Vzorec Gaussova filtru



Ilustrace 2: 2D tvar Gaussovy funkce, $\sigma = 2,5$

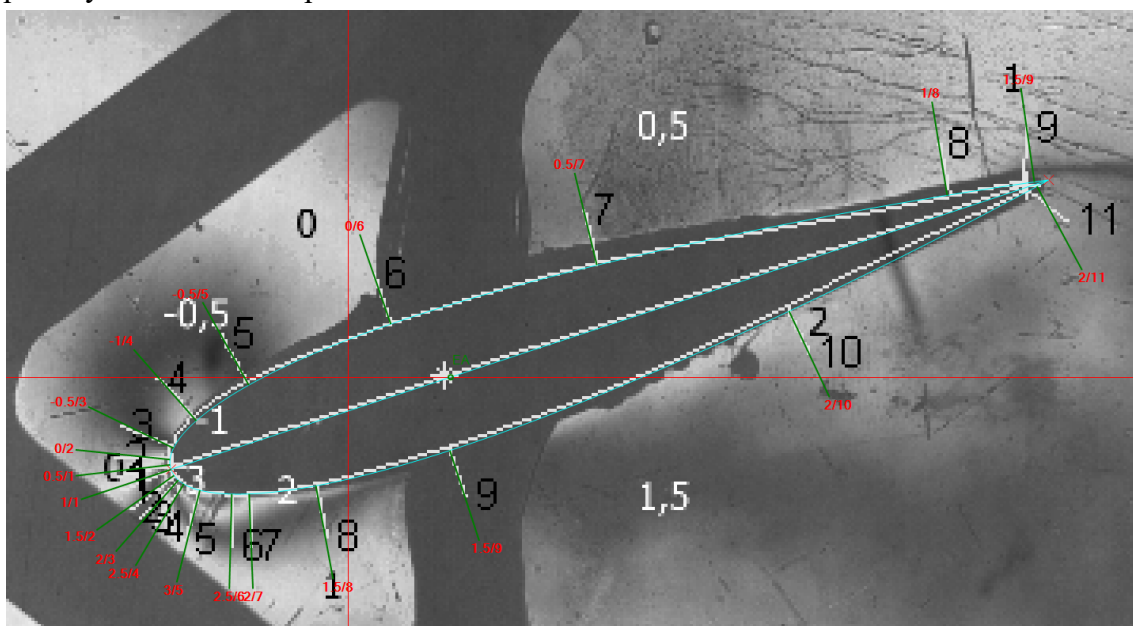
¹² Image processing, analysis, and machine vision. 4th edition. S.l.: Nelson Engineering, 2014. ISBN 1-133-59369-0. S. 140-141.

2 Použité prostředky

V této kapitole lze nalézt výčet použitých prostředků, technologií a jejich popis, kterých bylo potřeba k tvorbě aplikace.

2.1 IFGPro

IFGPro je program vytvořený programátorem Štěpánem Risseem za účelem zpracování dat z interferometrických měření. Po načtení obrazu je přesně určena pozice náběhové a odtokové hrany a rozměr modelu, se kterým se pracovalo, a pomocí toho je vykreslen profil modelu, na který jsou umisťovány ve specifických místech proužky, díky nimž se počítá vztaková a odporová síla a moment síly působící na profil. Program poté exportuje veškerá vypočtená data do textových souborů se všemi potřebnými informacemi o profilu v daném okamžiku.



Ilustrace 3: Vyhodnocení v programu IFGPro

2.2 Microsoft Visual Studio

Microsoft visual studio je IDE od Microsoftu, které je používáno pro vývoj konzolových aplikací, grafických aplikací, webových stránek a webových aplikací. Toto vývojové prostředí podporuje programovací jazyky prostřednictvím jazykových

služeb, díky čemuž může podporovat jakýkoliv jazyk. Mezi vestavěné jazyky patří C/C++, VB.NET a C#. Práce je zpracována v komunitní verzi Microsoft Visual Studio 2015.

2.3 Programovací jazyk C#

Je jazyk založený na jazycích C++ a Java, je tedy nepřímým potomkem jazyka C, z něhož bere syntaxi. Jedná se o vysokoúrovňový objektově orientovaný jazyk vyvinutý firmou Microsoft zároveň s platformou .NET Framework. Jedná se v dnešní době o jeden z nejmodernějších a nejpoužívanějších jazyků.

Tento jazyk byl zvolen z důvodu kompatibility, protože aplikace v IFGPro je psaná právě v tomto jazyce, a kvůli rozšiřujícím knihovnám (OpenCV/EmguCV) umožňujícím snadnější práci s obrazem.

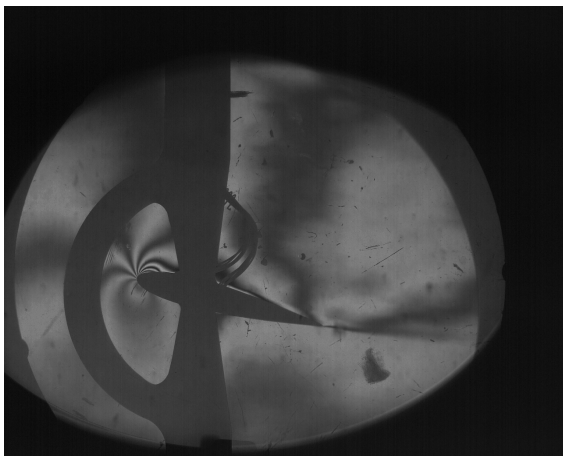
2.4 OpenCV

OpenCV (*Open Source Computer Vision*) je knihovna programových funkcí vyvinuta společností Intel a lze ji využít v jazycích C a C++. Je zaměřena především na počítačové vidění a zpracování obrazu v reálném čase.

2.5 EmguCV

Kvůli nekompatibilitě s jazykem C# musela být knihovna OpenCV importována pomocí EmguCV, který je multiplatformní .NET wrapper na knihovnu pro zpracování obrazu OpenCV. Pro zpřístupnění funkcí knihovny OpenCV musí být volán .NET kompatibilních jazyků, jako je C #, VB, VC ++, IronPython atd. NET wrapper může být zkompileovaný ve Visual Studio nebo jiných IDE a může fungovat na Windows, Linux, Mac OS X, iOS, Android a Windows Phone.

2.6 Snímky z měření



Ilustrace 4: Snímek z interferometru



Ilustrace 5: Snímek z měření

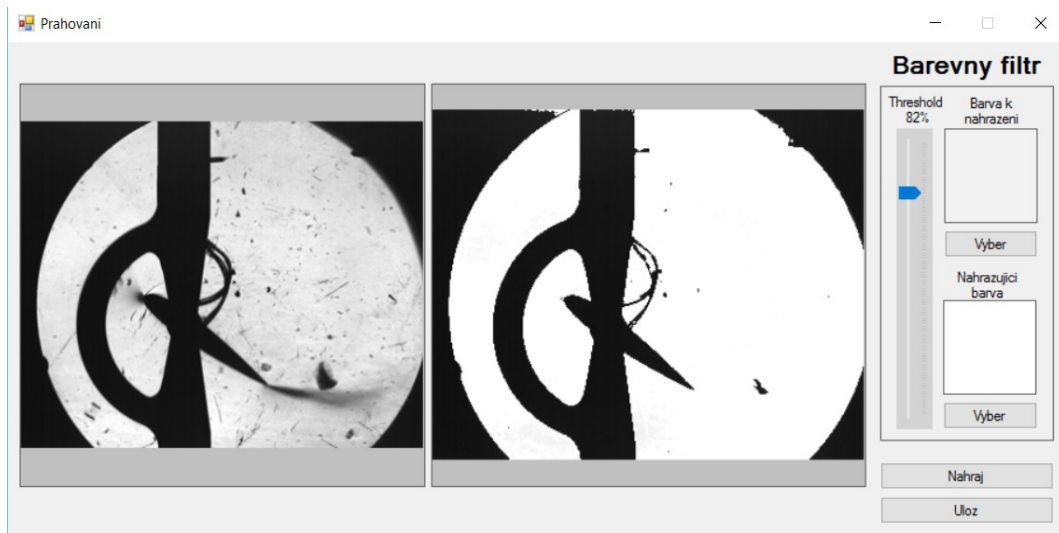
Snímky, které jsou vstupem do programu, vznikají při měřeních v Novém Kníně a jsou pořizovány vysokorychlostní kamerou. Snímky jsou pořizovány ve dvojicích, kterými jsou snímky z interferometru a snímky ze snímání světelných podmínek.¹³

¹³ M. Štěpán, P. Šidlof, V. Vlček. Měřicí řetězec aerodynamického profilu se dvěma stupni volnosti In: Interaction and feedbacks, national seminar with international participation, Prague: proceedings = Interakce a zpětné vazby, sborník referátů. Prague: Institute of Thermomechanics AS CR, 2013. S 3.

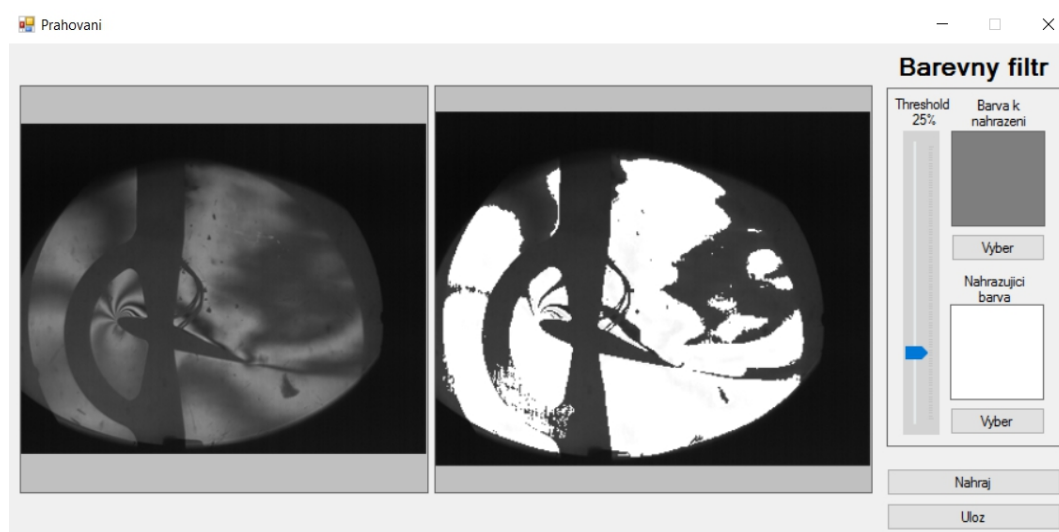
3 Návrhy řešení

V této kapitole jsou popsána různá řešení problému, která byla v průběhu práce použita. Po nastudování dostupných materiálů a typů snímků z měření jsem došel k závěru, že nejvhodnějším řešením problému bude použití hranových detektorů nebo prahování a segmentace obrazu. Z důvodu různého typu snímků bylo třeba nalézt postup, při kterém budou vyhodnocená data stejná na obou typech snímků. V následujících kapitolách jsou popsány používané operace na obou sadách používaných snímků.

3.1 Prahování



Ilustrace 6: Prahování fotografie vysokorychlostní kamery



Ilustrace 7: Prahování fotografie interferogramu

Tato metoda používá již zmíněné metody a postupy (viz. kapitola 1.1.1). Z ilustrací 6 a 7 je zřejmé, že metoda prahování nemohla být použita pro automatické nalezení náběhové a odtokové hrany. Na obou snímcích je jasně zřetelný aerodynamický profil, ale z důvodu použití různé hranice pro prahování a různých barev jednotlivých fotografií, by byla výsledná detekce hran v obraze chybná a nalezená náběhová a odtoková hrana leteckého profilu by neodpovídala skutečnosti.

3.2 Hranové detektory

Hranové detektory jsou v tomto případě nejlepším a nejefektivnějším způsobem nalezení hran v obraze a poté nalezení náběhové a odtokové hrany, protože pracují na podobných principech, které mají vyšší živočichové pro vnímání obrazu. Z těchto detektorů bylo potřeba vybrat ten, který bude mít nejpřesnější detekci celého profilu a náběhové a odtokové hrany, a dále ten, který bude dostatečně rychlý pro automatickou detekci a nebude ovlivněn různými typy snímků získaných z měření.



Ilustrace 8: Fotografie

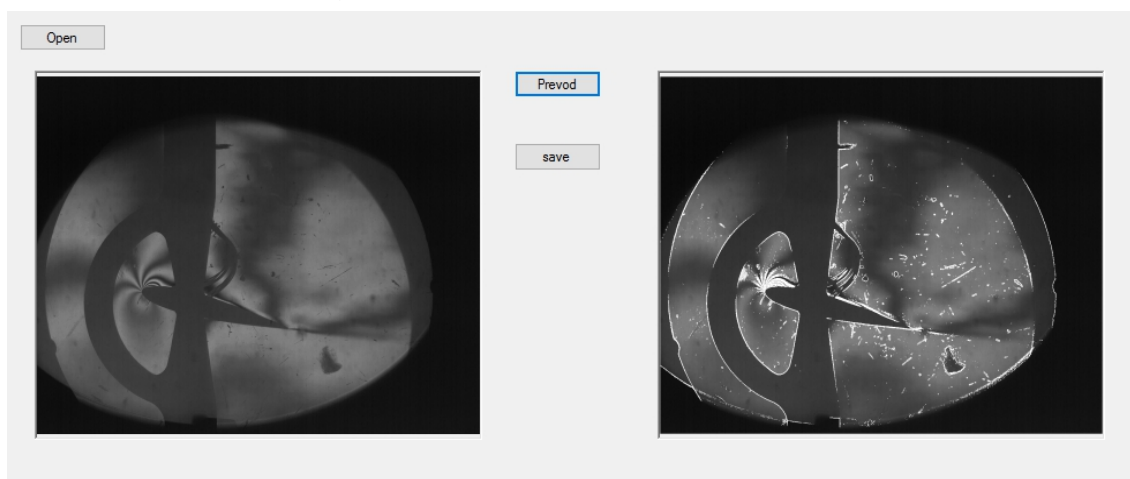
Ilustrace 9: Fotografie interferogramu

3.2.1 Maticové hranové detektory

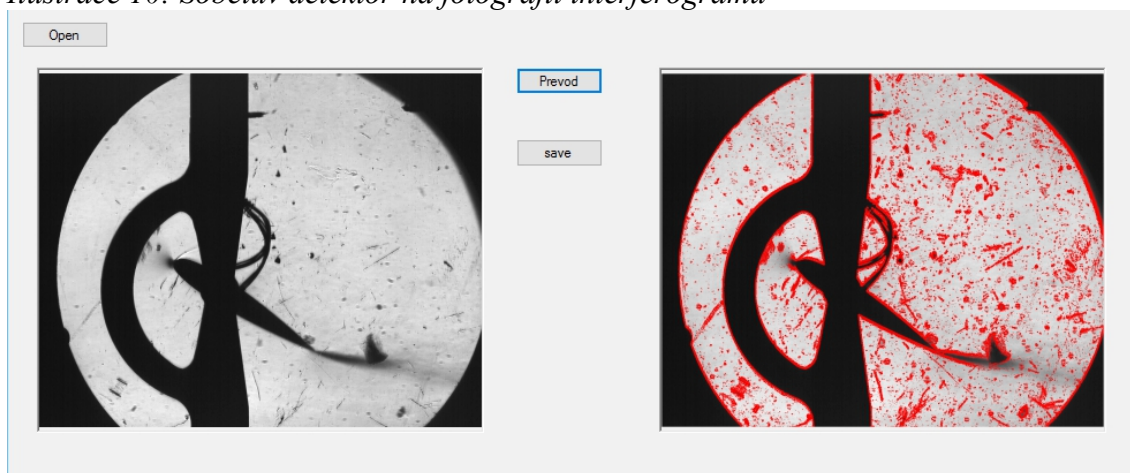
Tyto detektory používají metody a postupy (popsané v kapitole 1.2.1), díky nimž je velmi rychlý. Na ilustraci 8 můžete vidět, že při použití je oblast v okolí náběhové a odtokové hrany značně zkreslena, kvůli zachycení změn proudění vzduchu. Ilustrace 9

má již ostrou hranu u náběhové hrany, ale odtoková hrana je opět značně zkreslena a nemohla být tedy přesně nalezena. Z rozdílného chování snímků nemohl být tento postup použit pro automatické zpracování, protože by používané snímky vracely různé hodnoty.

3.2.2 Sobelův hranový detektor



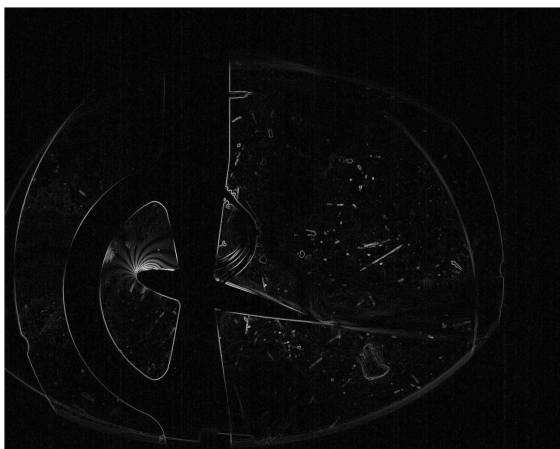
Ilustrace 10: Sobelův detektor na fotografii interferogramu



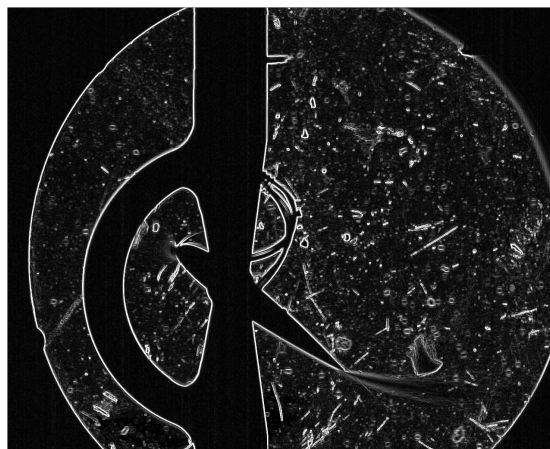
Ilustrace 11: Sobelův detektor na fotografii z vysokorychlostní kamery

Tento detektor používá již zmíněné metody a postupy (viz. kapitola 1.2.5). Na ilustracích 10 a 11 je zřejmé, že tento algoritmus je vhodný pro automatické zpracování, protože nalezené hrany jsou v okolí náběhové a odtokové hrany dostatečně ostré a nezkreslené. Tento algoritmus se mi ovšem nepodařilo zpracovat tak, aby byl dostatečně rychlý, a pro tento omezující faktor nebyl vybrán pro konečné zpracování, protože pokud by detekce leteckého profilu trvala příliš dlouho, nemělo by toto automatické hledání velký smysl.

3.2.3 Prewittův hranový detektor



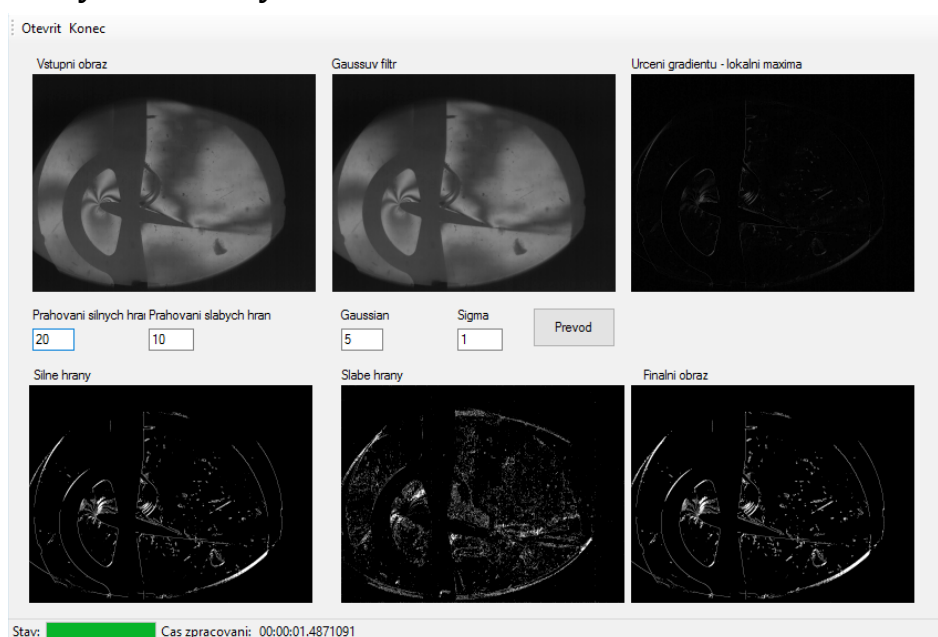
Ilustrace 12: Prewittův operátor



Ilustrace 13: Prewittův detektor

Tento hranový detektor (popsaný v kapitole 1.2.4) byl zkoušen pro detekci z důvodu jeho velké rychlosti. Ovšem na testových snímcích byl výsledný obraz značně rozmazán v okolí nalezených hran a bylo v něm velké množství nevýznamných detailů.

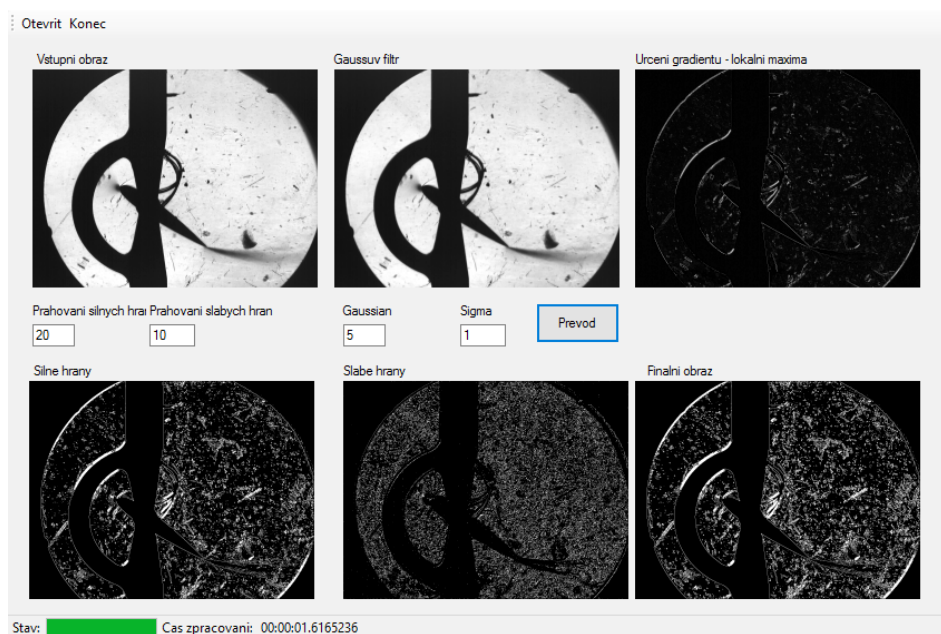
3.2.4 Cannyho hranový detektor



Ilustrace 14: Cannyho detektor na fotografii interferogramu

Tento algoritmus používá několik kroků popsanych v kapitole 1.2.6. Zpracování tohoto algoritmu je značně náročné z důvodu použití několika parametrů, a to velikosti Gaussianu a sigmy, ze kterých se počítá matice pro Gaussův filtr, který je poté

aplikován na snímek. Dalším vstupem do programu jsou hodnoty pro prahování, díky kterým jsou detekovány pouze skutečné hrany v leteckém profilu. Navzdory několika krokům, které tento algoritmus provádí, je na použitých snímcích značně rychlý a detekce trvá kolem 1.5 vteřiny, čímž je vhodný pro automatické zpracování. Tento hranový detektor nebývá ovlivněn různými typy snímků a při provedení vrací podobný obraz na všech typech snímků.



Ilustrace 15: Cannyho detektor na fotografii z vysokorychlostní kamery

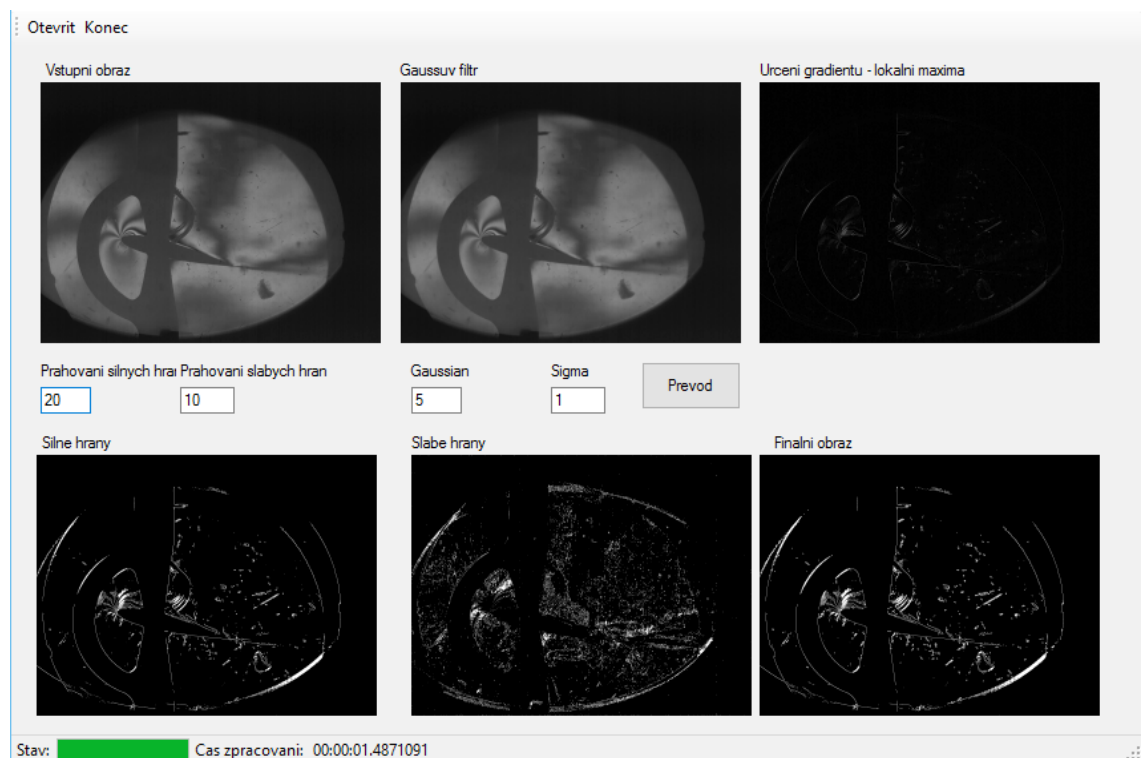
Při použití Cannyho hranového detektoru je důležité volit vhodné vstupní parametry pro přesně určení hran, protože jak můžete vidět na ilustraci 15, tak i Cannyho detektor detekuje při zvolení malých hodnot pro prahování nevýznamné detaily. Vstupní parametry byly zjišťovány postupným testováním na několika snímcích s různými hodnotami.

4 Realizace vlastního řešení

Po zpracování několika metod a postupů na zpracování obrazových snímků byl vybrán nejvhodnější hranový detektor, který je dostatečně rychlý a hrany detekuje bez výrazných šumů a nevýznamných detailů v obraze. Pro tento postup byl vybrán Cannyho hranový detektor, který pracuje v několika krocích a jeho jednotlivé části a následné nalezení náběhové a odtokové hrany jsou popsány v této kapitole.

4.1 Grafické rozhraní aplikace pro testování

Pro testování různých parametrů, které používá Cannyho hranový detektor, bylo vytvořeno grafické rozhraní, ve kterém jsou lehce viditelné všechny jednotlivé kroky tohoto detektoru pro snadnou manipulaci. Aplikace také disponuje ukládáním výsledného obrazu pro přesnější kontrolu, zda byla nalezena opravdu náběhová a odtoková hrana, a také stavovým panelem, který udává časový údaj potřebný pro zpracování jednoho snímku.¹⁴

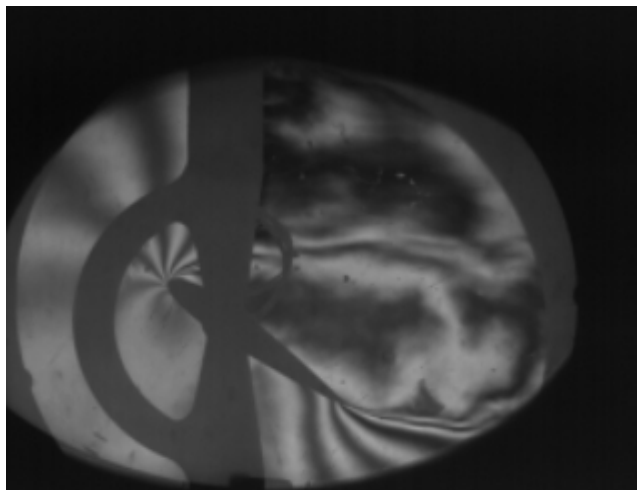


Ilustrace 16: Aplikace pro rozbor Cannyho detektoru

¹⁴ Nastudováno z: ALBAHARI, Joseph., Ben. ALBAHARI a Peter. DRAYTON. C# 5.0 in a nutshell. 5th ed. Sebastopol: O'Reilly, c2012. In a nutshell (O'Reilly & Associates). ISBN 9781449320102. S. 196

4.2 Vyhlazení: rozmazání obrazu pro odstranění šumu

Snímek je nejprve převeden do stupní šedi, což je první krok Cannyho detektoru. Jak je vidět na ilustraci 17, tak tento krok není u snímků z měření potřebný, protože snímky z interferometru jsou již ve stupních šedi a druhá série snímků je pouze černobílá. Je ale raději zahrnut do algoritmu pro možné budoucí zpracování i jiných snímků leteckých profilů.



Ilustrace 17: Úprava Gaussovým filterem

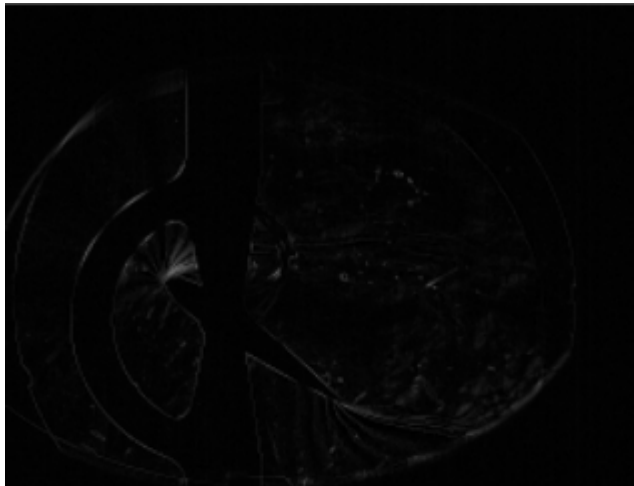
Při pořizování jakýchkoliv snímků je nevyhnutelné, že obraz bude obsahovat také určité množství šumu. Snímky z měření jsou dostatečně ostré, ale pro přesnější výsledky bylo vyhlazování také zahrnuto mezi kroky Cannyho detektoru. Tento krok lze upravit pomocí parametrů `Gaussian` a `Sigma`. Tyto parametry jsou dosazeny do funkce Gaussova filtru (viz kapitola 1.4.4) a je tím vypočtena matice pro následné zpracování.

Ukázka zdrojového kódu:

```
private int[,] GaussianFilter(int[,] Data){
    GenerateGaussianKernel(KernelSize, Sigma, out KernelWeight);
    int[,] Output = new int[Width, Height];
    int i, j, k, l;
    int Limit = KernelSize / 2;
    float Sum=0;
    Output = Data;
    for (i = Limit; i <= ((Width - 1) - Limit); i++){
        for (j = Limit; j <= ((Height - 1) - Limit); j++){
            Sum = 0;
            for (k = -Limit; k <= Limit; k++){
                for (l = -Limit; l <= Limit; l++){
                    Sum = Sum + ((float)Data[i + k, j + l] *
                    GaussianKernel [Limit + k, Limit + l]);
                }
            }
            Output[i, j] = (int) (Math.Round(Sum/ (float)KernelWeight));
        }
    }
    return Output;
}
```

4.3 Nalezení gradientů a lokálních maxim

Pro nalezení gradientů jsou použity matice Sobelova hranového detektoru popsané v kapitole 1.2.5, protože nejsou příliš citlivé na šum a vrací nejen směr gradientu, ale i velikost gradientu. Tyto matice jsou napsány přímo ve zdrojovém kódu, a nelze tedy tuto matici měnit za běhu aplikace.



Ilustrace 18: Nalezení gradientu a lokálních maxim

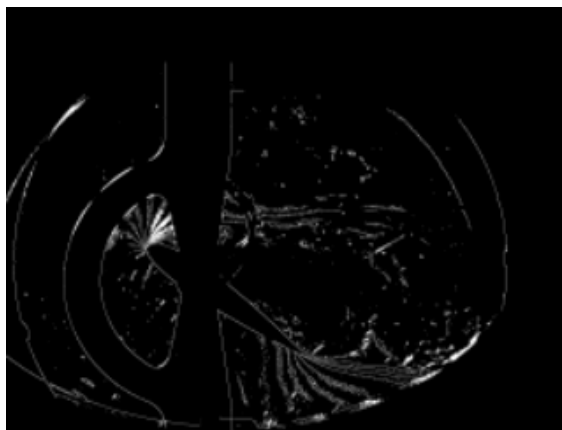
Při nalezení lokálních maxim jsou použity gradienty z předchozího kroku, úhly směru působení gradientu jsou nejprve převedeny z radiánů na stupně velikosti úhlu, a poté jsou procházeny ve směrech horizontálních a vertikálních hran a dále ve směru úhlu 45° a 135° . Nalezené pixely jsou ty, jejichž okolí je ve směru a proti směru gradientu nižší. Směry gradientu jsou přesně stanoveny ve zdrojovém kódu. Tento krok popisuje Ilustrace 18.

Ukázka zdrojového kódu hledání hrany v horizontálním směru:

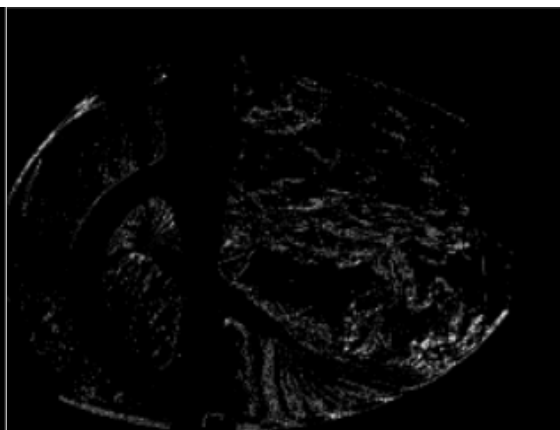
```
int r, c;
float Tangent;
for (i = Limit; i <= (Width - Limit) - 1; i++){
    for (j = Limit; j <= (Height - Limit) - 1; j++){
        if (DerivativeX[i, j] == 0)
            Tangent = 90F;
        else
            Tangent = (float)(Math.Atan(DerivativeY[i, j] / DerivativeX[i,
j]) * 180 / Math.PI);
        //Horizontalni hrana
        if (((- 2.5 < Tangent) && (Tangent <= 22.5)) || ((157.5 < Tangent)
&& (Tangent <= -157.5))){
            if ((Gradient[i, j] < Gradient[i, j + 1]) || (Gradient[i, j] <
Gradient[i, j - 1]))
                NonMax[i, j] = 0;
        }
    }
}
```


4.4 Eliminace nevýznamných hran – prahování

Při eliminaci nevýznamných hran je použit princip prahování popsaný v kapitole 1.1.1. Cannyho detektor používá dvě prahové hodnoty pro potlačení krátkých/slabých bezvýznamných hran. Při detekování hrany jsou silné hrany prahovány hodnotou 60, což vede k tomu, že zůstanou detekované hrany pouze leteckého profilu, a hrany vlnění v okolí náběhové a odtokové hrany a slabé hrany hodnotou 50, díky čemuž jsou zahrnuty i detaily v okolí náběhové a odtokové hrany.



Ilustrace 19: Určení silných hran



Ilustrace 20: Určení slabých hran

Ukázka zdrojového kódu prahování:

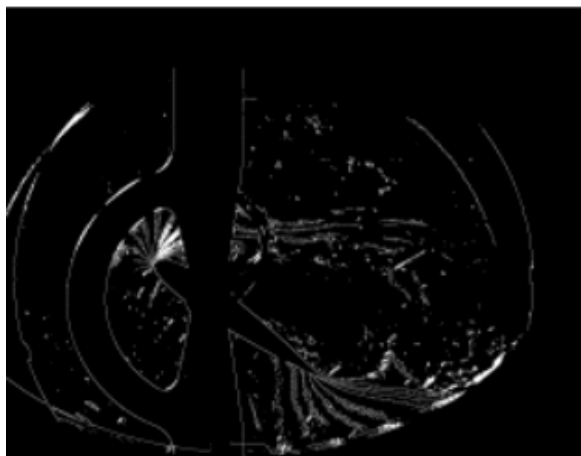
```
private void HysteresisThresholding(int[,] Edges)
{
    int i, j;
    int Limit= KernelSize/2;
    for (i = Limit; i <= (Width - 1) - Limit; i++){
        for (j = Limit; j <= (Height - 1) - Limit; j++){
            if (Edges[i, j] == 1){
                EdgeMap[i, j] = 1;
            }
        }
    }

    for (i = Limit; i <= (Width - 1) - Limit; i++){
        for (j = Limit; j <= (Height - 1) - Limit; j++){
            if (Edges[i, j] == 1){
                EdgeMap[i, j] = 1;
                Travers(i, j);
                VisitedMap[i, j] = 1;
            }
        }
    }
    return;
}
```

Hodnoty pro prahování byly takto zvoleny dlouhým testováním a jsou zadány přímo do zdrojového kódu a nelze je měnit za běhu aplikace, protože po změně těchto hodnot by se ve výsledném obraze mohly objevit i nepotřebné detaily a finální detekce hrany by proto mohla být značně zkreslena.

4.5 Sledování hranice hysterezí

Všechny výsledné hrany jsou určeny potlačením slabých hran, které nejsou připojeny k nějaké silné hraně. Tohoto je dosaženo rekurzivní funkcí, která prahuje obraz dvěma parametry (popsáno v předchozím kroku) a používá propojení osmi sousedních pixelů pro určení, zda slabá hrana náleží k nějaké silné.



Ilustrace 21: Výsledný obraz

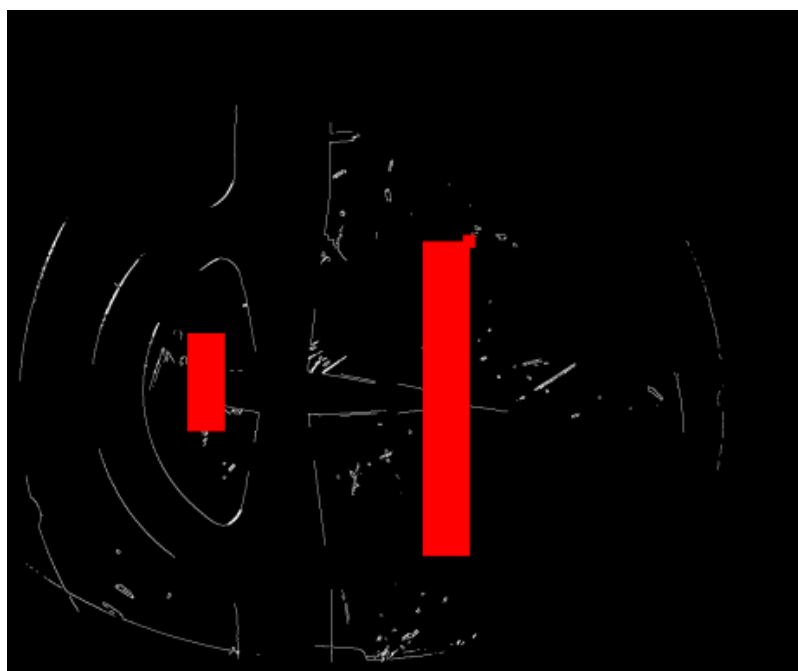
Výsledný obraz je spojení silných hran určených prahování a slabých hran, které jsou určeny hysterezí. Na výsledném obraze můžete vidět, že hrany jsou detekovány v okolí leteckého profilu a u náběhové a odtokové hrany, které jsou způsobeny vlněním vzduchu.

Ukázka zdrojového kódu hledání sousedních pixelů:

```
//1
if (EdgePoints[X + 1, Y] == 2)
{
    EdgeMap[X + 1, Y] = 1;
    VisitedMap[X + 1, Y] = 1;
    Travers(X + 1, Y);
    return;
}
```


4.6 Nalezení náběhové a odtokové hrany

Poté, co proběhne Cannyho hranový detektor jsou hrany označeny bíle. V obraze jsou ovšem také různé nevýznamné detaily, které nejsou při hledání náběhové a odtokové hrany potřebné, proto je potřeba hledat pouze oblasti s pixely, které odpovídají náběhové a odtokové hraně, a při porovnávání jsou prohledávány pouze oblasti v okolí náběhové a odtokové hrany. Prohledávaná oblast u náběhové hrany je omezena na ose x mezi pixely 290-350 a 500-650 na ose y, oblast u odtokové hrany je omezena na ose x mezi pixely 665 – 740 a 360 – 840 na ose y (viz Ilustrace 22).



Ilustrace 22: Prohledávané oblasti

Při porovnávání je brán ohled na to, aby byly porovnávány pouze pixely na krajích leteckého profilu a tedy z jedné strany mají velké množství černých pixelů. Protože pokud by tomu tak nebylo, mohly by být místo náběhové a odtokové hrany určeny některé nevýznamné detaily.

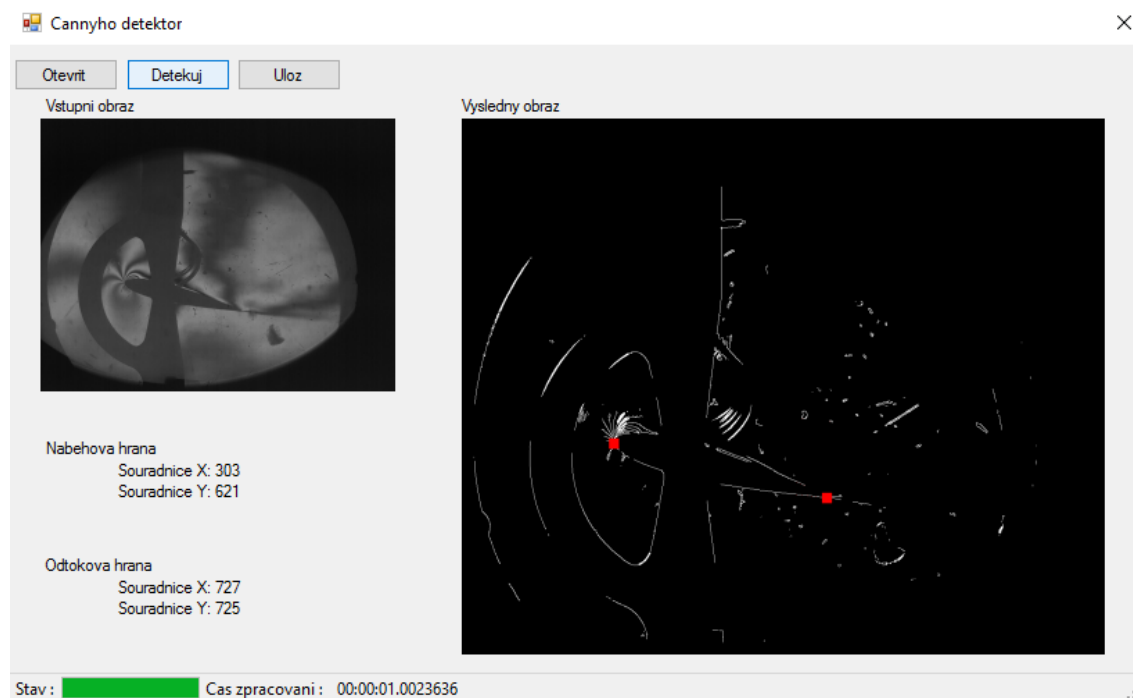
Algoritmus byl upraven, aby u porovnávaných pixelů byly sledovány také různé vlastnosti, které byly zjištěny opakovanou detekcí snímků v krajních polohách a podle kterých se chovají pixely u náběhové a odtokové hrany. Pomocí těchto objevených opakujících se chování byl algoritmus upraven, aby byl co nejpřesnější.

Po nalezení přesné souřadnice náběhové a odtokové hrany v pixelech od levého horního rohu jsou tyto údaje zobrazeny na pozadí aplikace a pro jasnou vizualizaci je okolí těchto bodů obarveno červenou barvou.

Ukázka zdrojového kódu nalezení náběhové hrany:

```
for (int k = 290; k < 350; k++){ //sirka
    for (int l = 500; l < 650; l++){ //vyska
        if (ret.GetPixel(k,l).ToArgb()==Color.White.ToArgb()
            && ret.GetPixel(k + 1, l).ToArgb() == Color.Black.ToArgb()
            && ret.GetPixel(k + 2, l).ToArgb() == Color.Black.ToArgb()
            && ret.GetPixel(k + 3, l).ToArgb() == Color.Black.ToArgb()
            && ret.GetPixel(k + 4, l).ToArgb() == Color.Black.ToArgb()
            && ret.GetPixel(k + 5, l).ToArgb() == Color.Black.ToArgb()
            && ret.GetPixel(k, l + 1).ToArgb() == Color.White.ToArgb()
            && ret.GetPixel(k, l - 1).ToArgb() == Color.White.ToArgb()
            && ret.GetPixel(k, l - 2).ToArgb() == Color.White.ToArgb()
            && ret.GetPixel(k, l - 3).ToArgb() == Color.White.ToArgb()
            && ret.GetPixel(k - 1, l).ToArgb() == Color.White.ToArgb())
        {
            ret.SetPixel(k, l, Color.Red);
            if (k < nabehXmax && l < nabehYmax){
                nabehXmax = k;
                nabehYmax = l;
            }
        }
    }
}
```

4.7 Testování



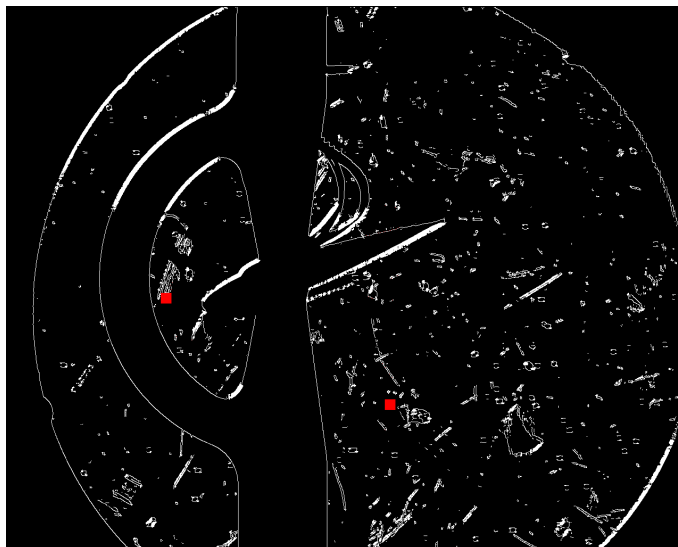
Ilustrace 23: Výsledný vzhled aplikace

Nedílnou součástí realizace algoritmu bylo také otestování tohoto algoritmu na co největším počtu snímků z důvodu jejich dvou různých typů. Vzhledem ke dvěma různým sadám snímků a jejich obrovskému množství je téměř nemožné otestovat tento algoritmus na všech snímcích z měření.

Při testování byl algoritmus stále upravován, díky podobnosti snímaných obrazů, pro stále přesnější detekci polohy, a také aby byl odolnější a stále fungoval na snímcích, na kterých ještě testován nebyl. S detekcí není ve většině případů problém, občas se ale vyskytnou jednotlivé snímky, kde detekce poskytne lehce nepřesné výsledky, chyba je ale v řádech jednotek pixelů, ve zcela ojedinělých případech detekce zcela selhala.

Při testování snímků z měření bylo zjištěno, že druhá sada snímků má jiné rozměry a náběhová a odtoková hrana profilu se nachází úplně v jiné oblasti. Chování snímků v krajních pozicích (profil naprosto rovně, náběhová hrana v nejvyšším bodě a odtoková hrana v nejvyšším bodě) bylo také naprosto rozdílné od snímků z interferometru.

Protože zachycené rozdílné prodění vzduchu zabarvuje oblasti s vysokou hustotou dočerna, čímž může rozšiřovat letecký profil, není náběhová nebo odtoková hrana nalezena přesně viz - Ilustrace 24.



Ilustrace 24: Cannyho detektor na fotografii z vysokorychlostní kamery

Po konzultaci s vedoucím práce jsme se rozhodli, že detekci náběhové a odtokové hrany budeme provádět u snímků z interferometrických měření, protože právě ty jsou vstupním obrazem, které zpracovává program IFGPro.

Pokud by bylo požadováno zpracování sady snímků z vysokorychlostní kamery, tak nalezená náběhová a odtoková hrana nebude odpovídat, ale letecký profil bude stále obarven, protože se na něm hrana nalézá. Tohoto faktoru lze využít u obou sad snímků pro celkovou kontrolu, zda je letecký profil správně vykreslen v programu IFGPro a výsledky jsou tedy správné.

Závěr

Aplikace, vytvořená v rámci této práce, je schopna detekovat různé aerodynamické profily ve snímcích za předpokladu, že se daný aerodynamický profil bude nacházet ve středu snímku. K dispozici je jednoduché uživatelské rozhraní pro načítání snímků a následné vyhodnocení. Pro okamžité potvrzení správnosti nalezených hran je k dispozici náhled výsledného snímku, na kterém jsou zabarveny náběhová a odtoková hrana. Souřadnice náběhové a odtokové hrany jsou určeny pozicí pixelů na ose x a y od levého horního rohu. Výsledné obrazy je také samozřejmě možné uložit do běžných formátů používaných pro obrazové materiály jako jsou BMP, PNG a JPG.

Přesnost detekce náběhové a odtokové hrany je dostatečná pro další zpracování v programu IFGPro, ale v některých případech může být nalezená náběhová hrana mírně posunuta, či není detekována vůbec z důvodu velkých odlišností v krajních pozicích leteckého profilu ve snímcích. Z tohoto důvodu by nemělo být automatické detekování náběhové a odtokové hrany jediným způsobem nalezení hrany, ale do vyhodnocování by měl být také zahrnut lidský faktor pro korekci vykresleného profilu v programu IFGPro.

Tato aplikace, respektive její třídy, lze snadno implementovat do programu IFGPro pro snadnější práci, nebo ji použít jako extrení .dll knihovnu, kterou bude aplikace IFGPro pouze importovat. Pokud by aplikace měla fungovat samostatně, bylo by vhodné předělat GUI pro snadnější manipulaci s daty. Zlepšení by si v tomto případě zasloužilo také detekce hran na více typech snímků, či ruční zvolení použitého detektoru.

Seznam ilustrací

Funkce prahování.....	13
2D tvar Gaussovy funkce, $\sigma = 2,5$	19
Vyhodnocení v programu IFGPro.....	20
Snímek z interferometru.....	22
Snímek z měření.....	22
Prahování fotografie vysokorychlostní kamery.....	23
Prahování fotografie interferogramu.....	23
Fotografie.....	24
Fotografie interferogramu.....	24
Sobelův detektor na fotografii interferogramu.....	25
Sobelův detektor na fotografii z vysokorychlostní kamery.....	25
Prewittův operátor.....	26
Prewittův detektor.....	26
Cannyho detektor na fotografii interferogramu.....	26
Cannyho detektor na fotografii z vysokorychlostní kamery.....	27
Aplikace pro rozbor Cannyho detektoru.....	28
Úprava Gaussovým filtrem.....	29
Nalezení gradientu a lokálních maxim.....	30
Určení silných hran.....	31
Určení slabých hran.....	31
Výsledný obraz.....	32
Prohledávané oblasti.....	33
Výsledný vzhled aplikace.....	35
Cannyho detektor na fotografii z vysokorychlostní kamery.....	36

Seznam použité literatury

- [1] ŠIDLOF, Petr, Václav VLČEK, Martin ŠTĚPÁN, Jaromír HORÁČEK, Martin LUXA, David ŠIMURDA a Jan KOZÁNEK. Wind Tunnel Measurements of Flow-Induced Vibration of a NACA0015 Airfoil Model. In: Volume 4: Fluid-Structure Interaction [online]. 2014 [cit. 2015-05-03]. DOI: 10.1115/pvp2014-28294.
- [2] Štěpán M., Šidlof P., Vlček V. (2013), Měřicí řetězec aerodynamického profilu se dvěma stupni volnosti, in: *Interactions and feedbacks 2013*, Prague, Czech Republic.
- [3] Šonka V., Hlaváč M., Boyle R. (2014), Image Processing, Analysis, and Machine Vision, 4th edition.
- [4] Albahari J., Albahari B. (2012), C# 5.0 in a Nutshell: The Definitive Reference, 5th Edition.
- [5] Tropea C., Yarin A. L., (Eds.) J. F. F. (2007), Springer Handbook of Experimental Fluid Mechanics, Springer.
- [6] Shapiro A. H. (1954), The Dynamics and Thermodynamics of Compressible Fluid Flow, The Ronald Press Company, New York.